FIG. 1a

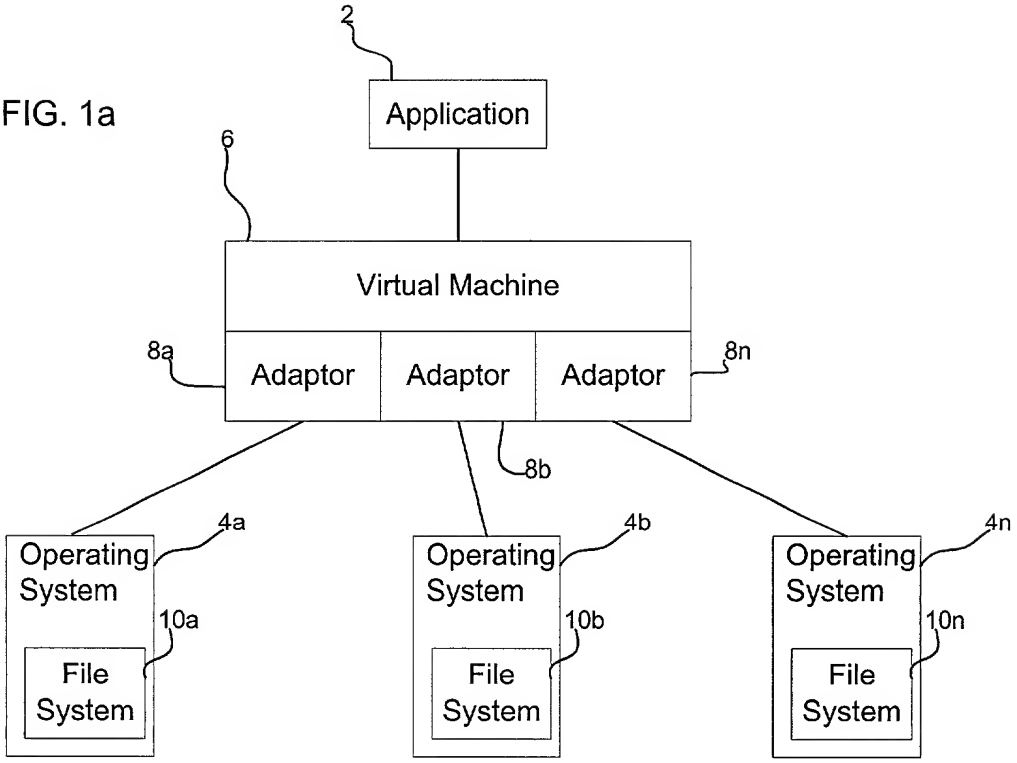

2
Application
6
Virtual Machine
8a
Adaptor
Adaptor
Adaptor
8n
8b
Operating System
4a
10a
File System
Operating System
4b
10b
File System
Operating System
4n
10n
File System

FIG. 1b

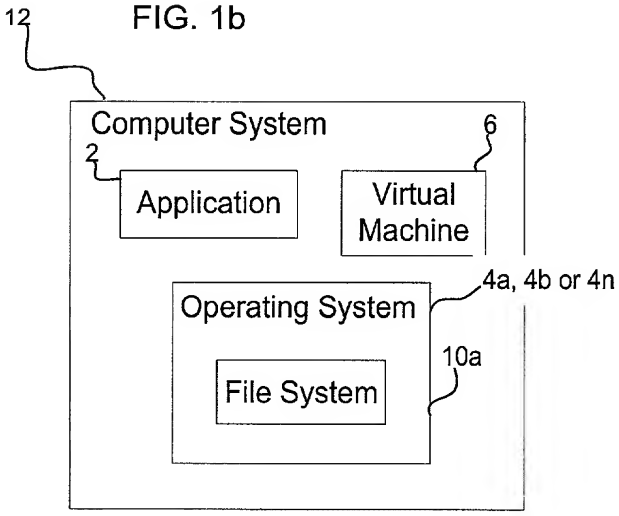

12
Computer System
6
2
Application
Virtual Machine
4a, 4b or 4n
Operating System
10a
File System

FIG. 2

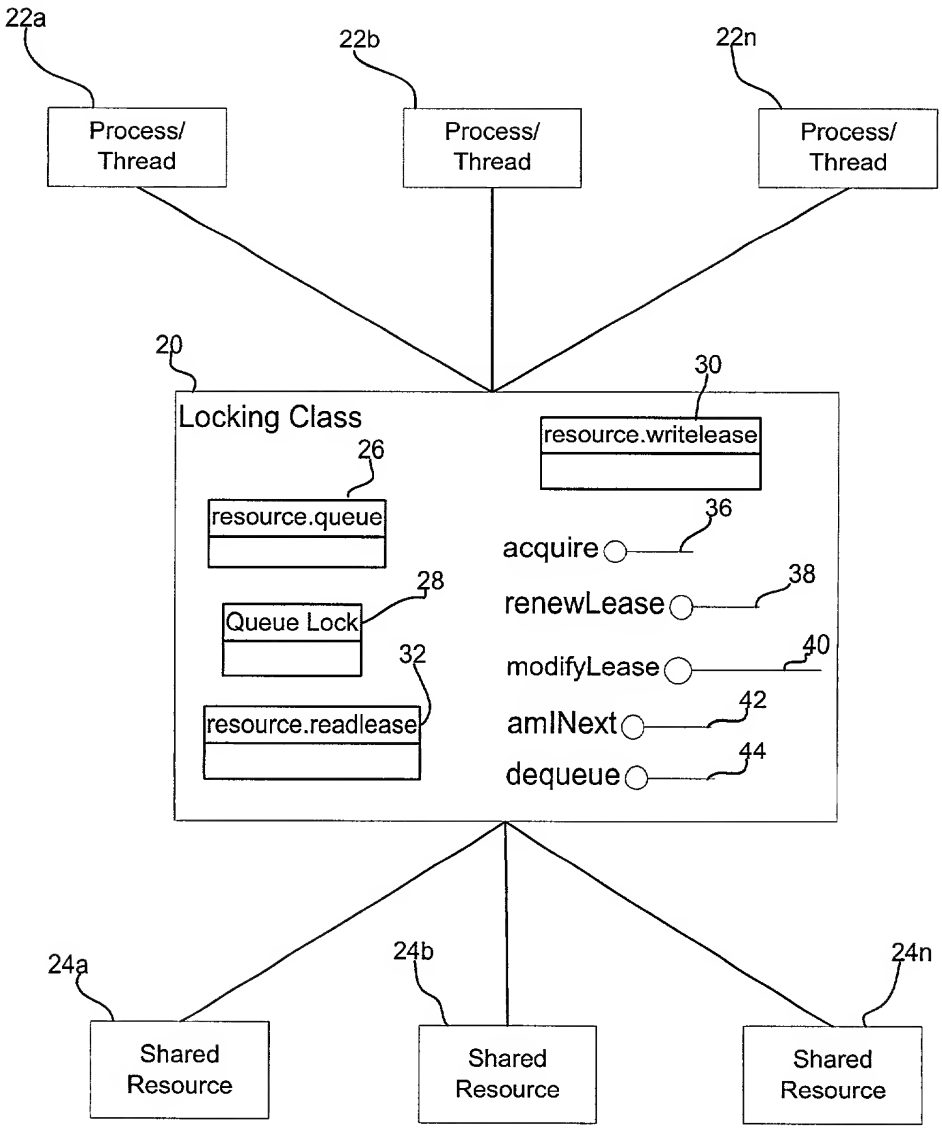

22a
22b
22n
Process/ Thread
Process/ Thread
Process/ Thread
20
Locking Class
30
resource.writelease
26
resource.queue
36
acquire
38
renewLease
28
Queue Lock
40
modifyLease
32
resource.readlease
42
amINext
44
dequeue
24a
24b
24n
Shared Resource
Shared Resource
Shared Resource

FIG. 3

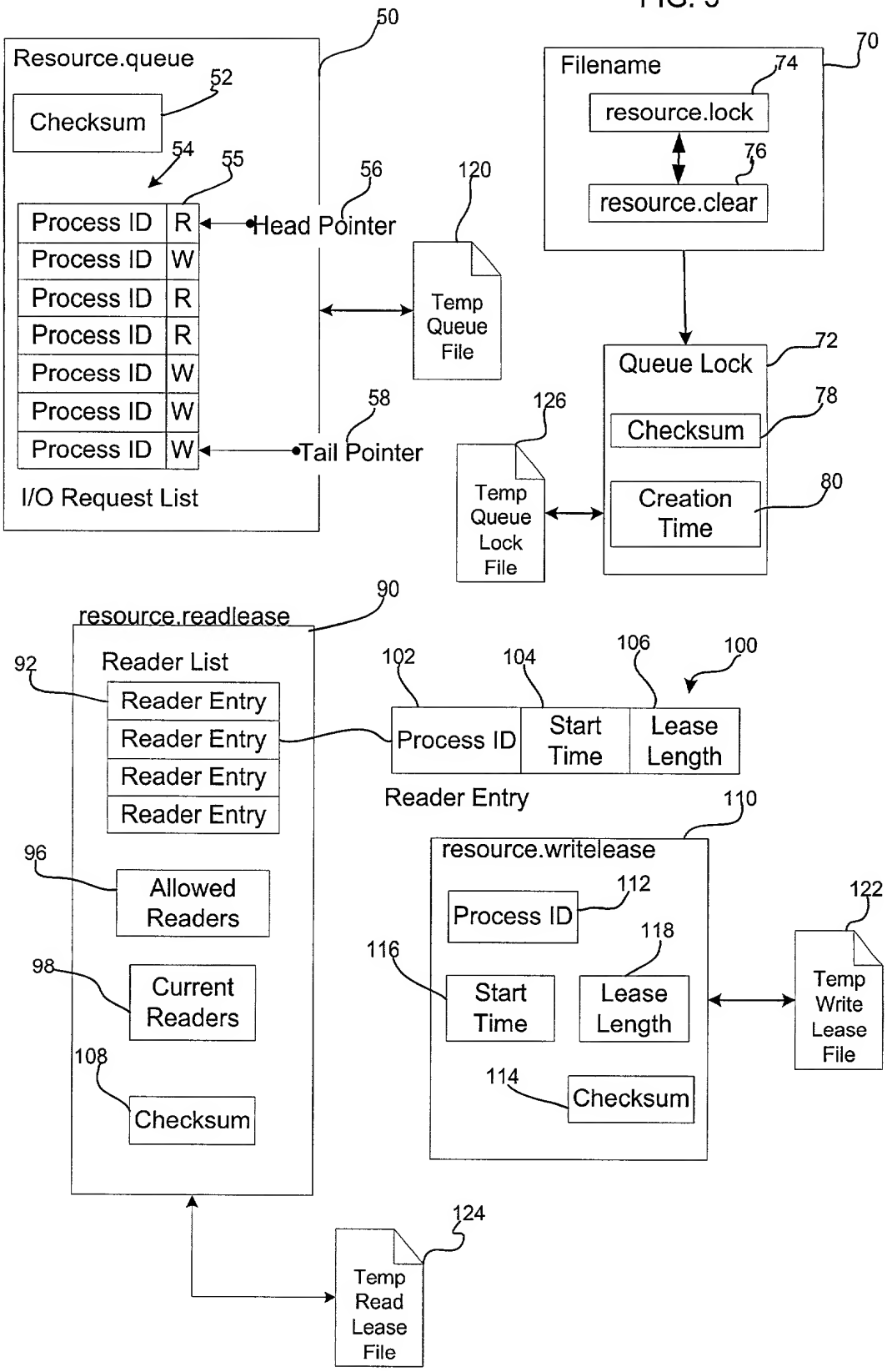
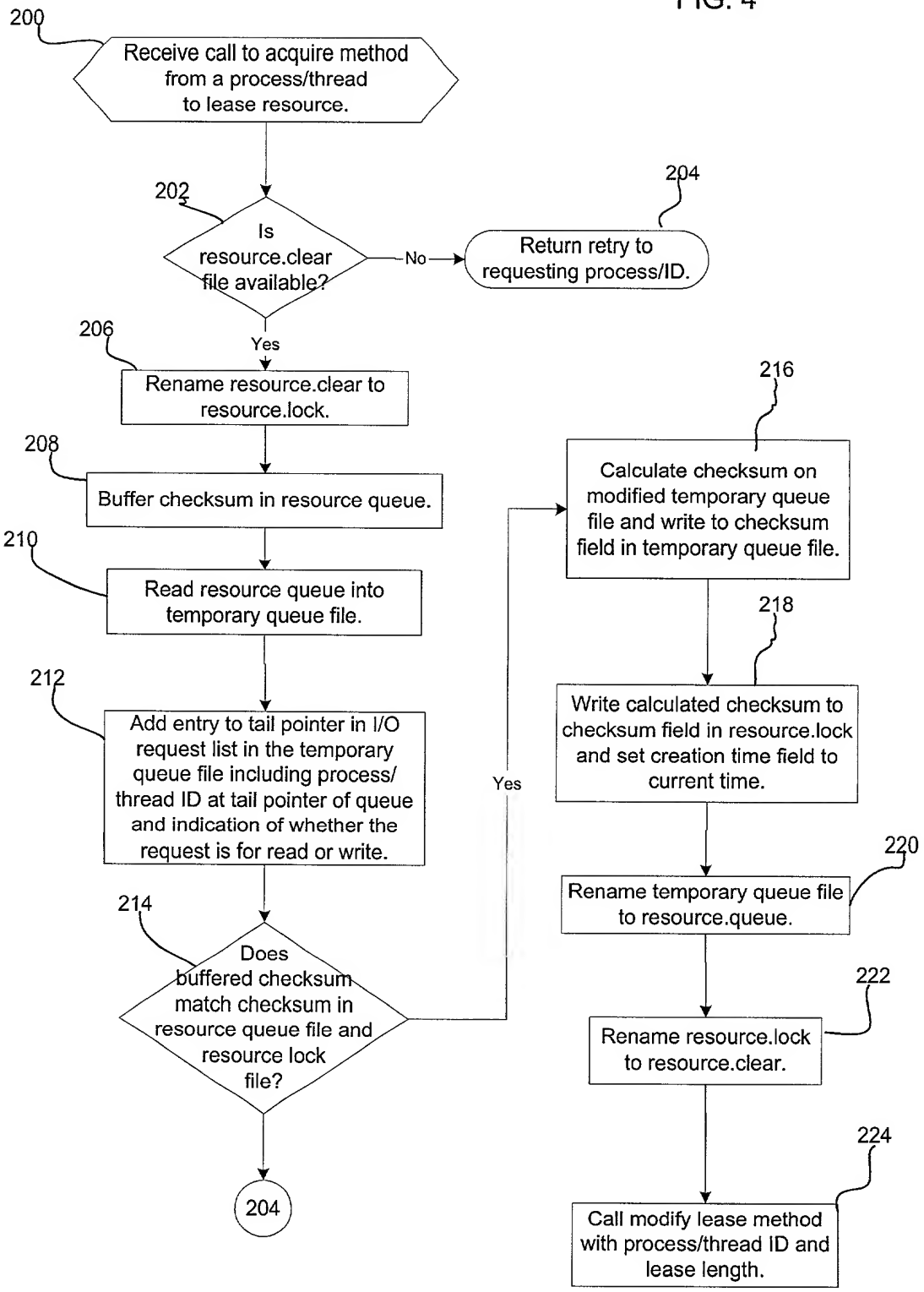

50
Resource.queue
52
Checksum
54
55
56
120
Process ID
R
Head Pointer
Process ID
W
Process ID
R
Process ID
R
Process ID
W
Process ID
W
Process ID
W
58
Tail Pointer
I/O Request List
Temp Queue File
70
Filename
74
resource.lock
76
resource.clear
72
Queue Lock
78
Checksum
126
80
Temp Queue Lock File
Creation Time
90
resource.readlease
92
Reader List
Reader Entry
Reader Entry
Reader Entry
Reader Entry
102
104
106
100
Process ID
Start Time
Lease Length
Reader Entry
110
resource.writelease
112
Process ID
122
116
118
Start Time
Lease Length
Temp Write Lease File
114
Checksum
96
Allowed Readers
98
Current Readers
108
Checksum
124
Temp Read Lease File

FIG. 4


200
Receive call to acquire method from a process/thread to lease resource.
202
Is resource.clear file available?
No
204
Return retry to requesting process/ID.
Yes
206
Rename resource.clear to resource.lock.
208
Buffer checksum in resource queue.
210
Read resource queue into temporary queue file.
212
Add entry to tail pointer in I/O request list in the temporary queue file including process/ thread ID at tail pointer of queue and indication of whether the request is for read or write.
214
Does buffered checksum match checksum in resource queue file and resource lock file?
Yes
204
216
Calculate checksum on modified temporary queue file and write to checksum field in temporary queue file.
218
Write calculated checksum to checksum field in resource.lock and set creation time field to current time.
220
Rename temporary queue file to resource.queue.
222
Rename resource.lock to resource.clear.
224
Call modify lease method with process/thread ID and lease length.

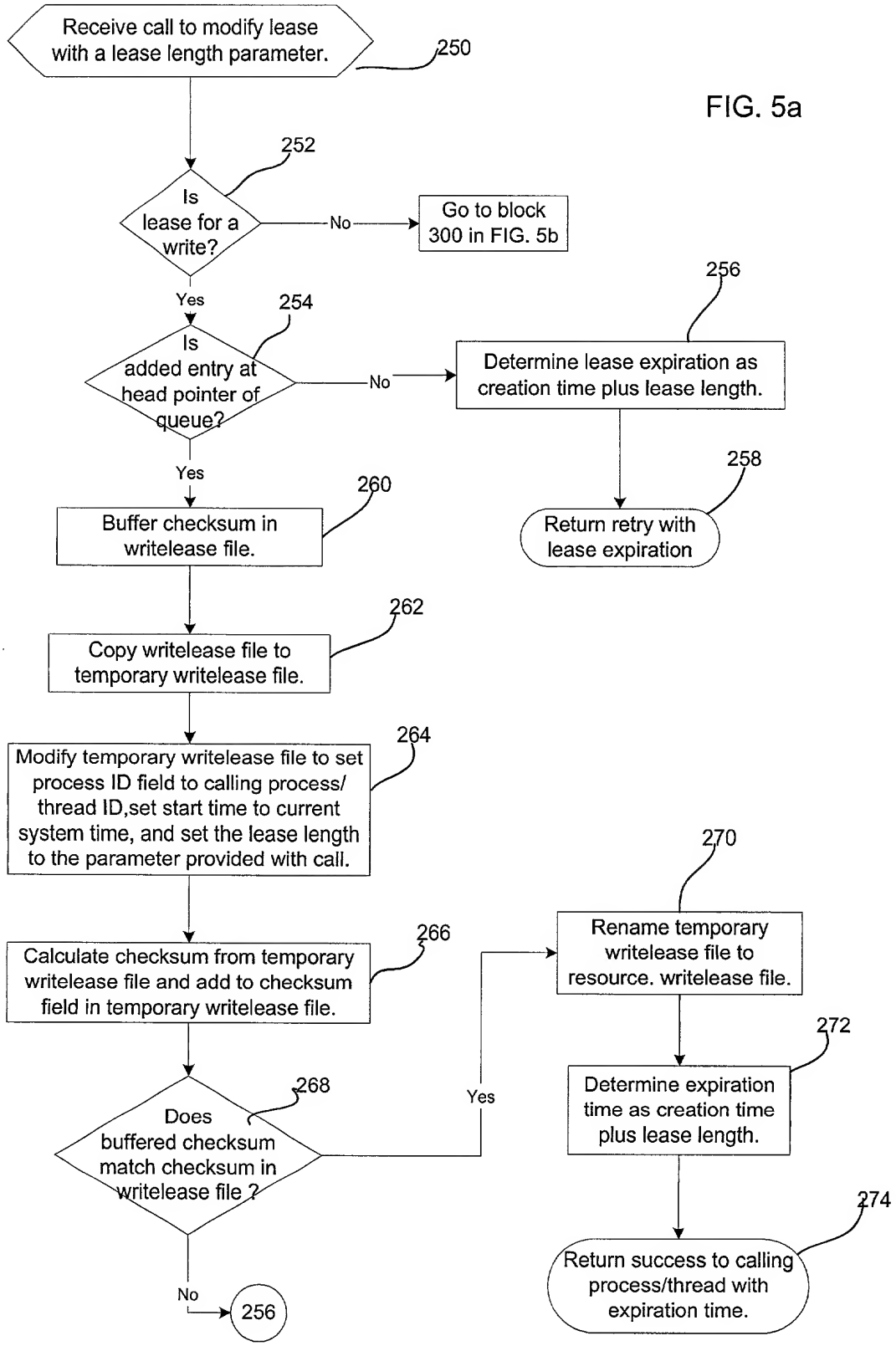
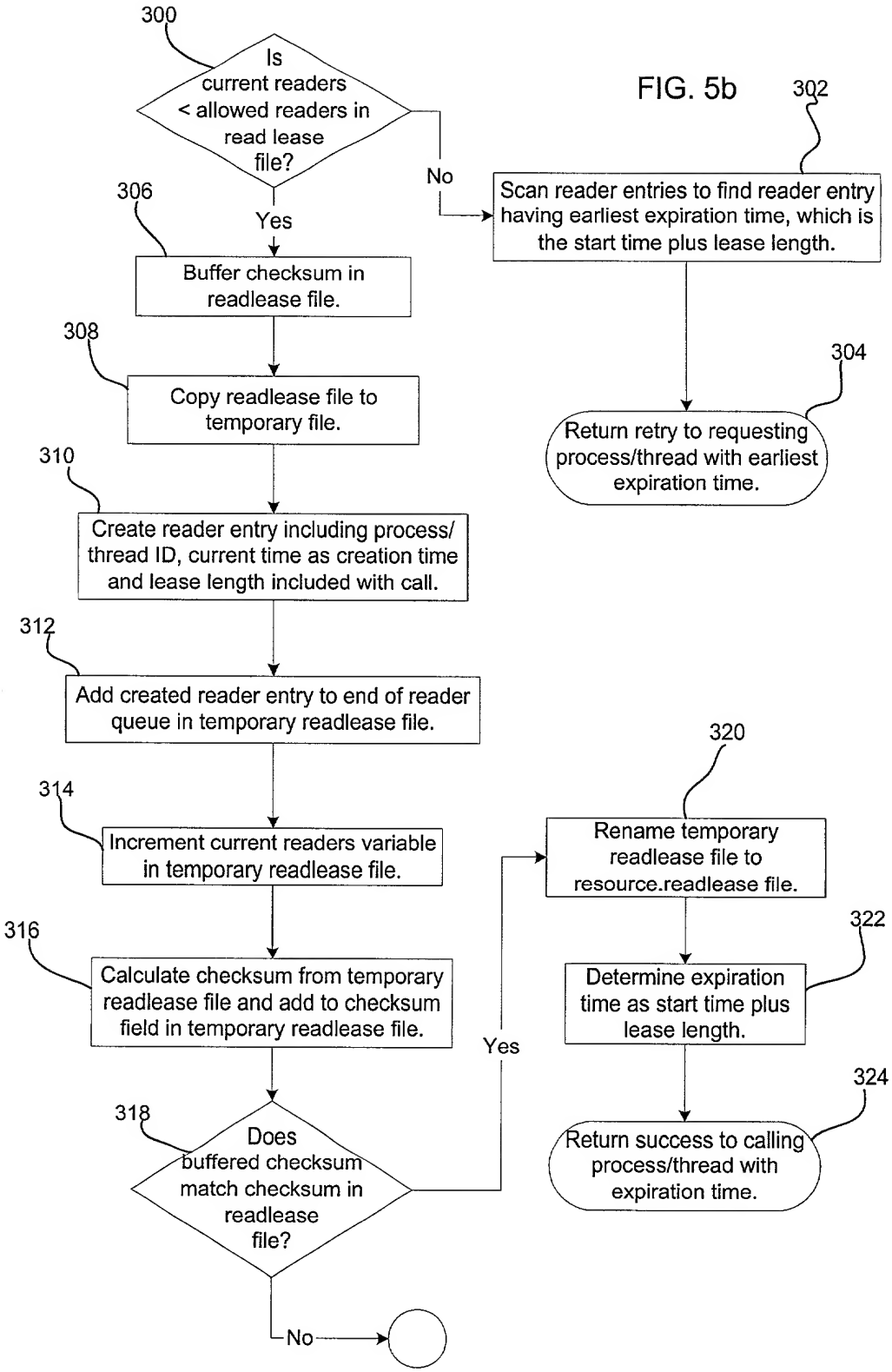

Receive call to modify lease with a lease length parameter.
250
252
Is lease for a write?
No
Go to block 300 in FIG. 5b
Yes
254
Is added entry at head pointer of queue?
No
256
Determine lease expiration as creation time plus lease length.
258
Return retry with lease expiration
Yes
260
Buffer checksum in writelease file.
262
Copy writelease file to temporary writelease file.
264
Modify temporary writelease file to set process ID field to calling process/ thread ID,set start time to current system time, and set the lease length to the parameter provided with call.
266
Calculate checksum from temporary writelease file and add to checksum field in temporary writelease file.
268
Does buffered checksum match checksum in writelease file ?
Yes
270
Rename temporary writelease file to resource. writelease file.
272
Determine expiration time as creation time plus lease length.
274
Return success to calling process/thread with expiration time.
No
256

FIG. 5a

FIG. 5b


300
Is current readers < allowed readers in read lease file?
No
Yes
302
Scan reader entries to find reader entry having earliest expiration time, which is the start time plus lease length.
304
Return retry to requesting process/thread with earliest expiration time.
306
Buffer checksum in readlease file.
308
Copy readlease file to temporary file.
310
Create reader entry including process/ thread ID, current time as creation time and lease length included with call.
312
Add created reader entry to end of reader queue in temporary readlease file.
314
Increment current readers variable in temporary readlease file.
316
Calculate checksum from temporary readlease file and add to checksum field in temporary readlease file.
318
Does buffered checksum match checksum in readlease file?
Yes
No
320
Rename temporary readlease file to resource.readlease file.
322
Determine expiration time as start time plus lease length.
324
Return success to calling process/thread with expiration time.

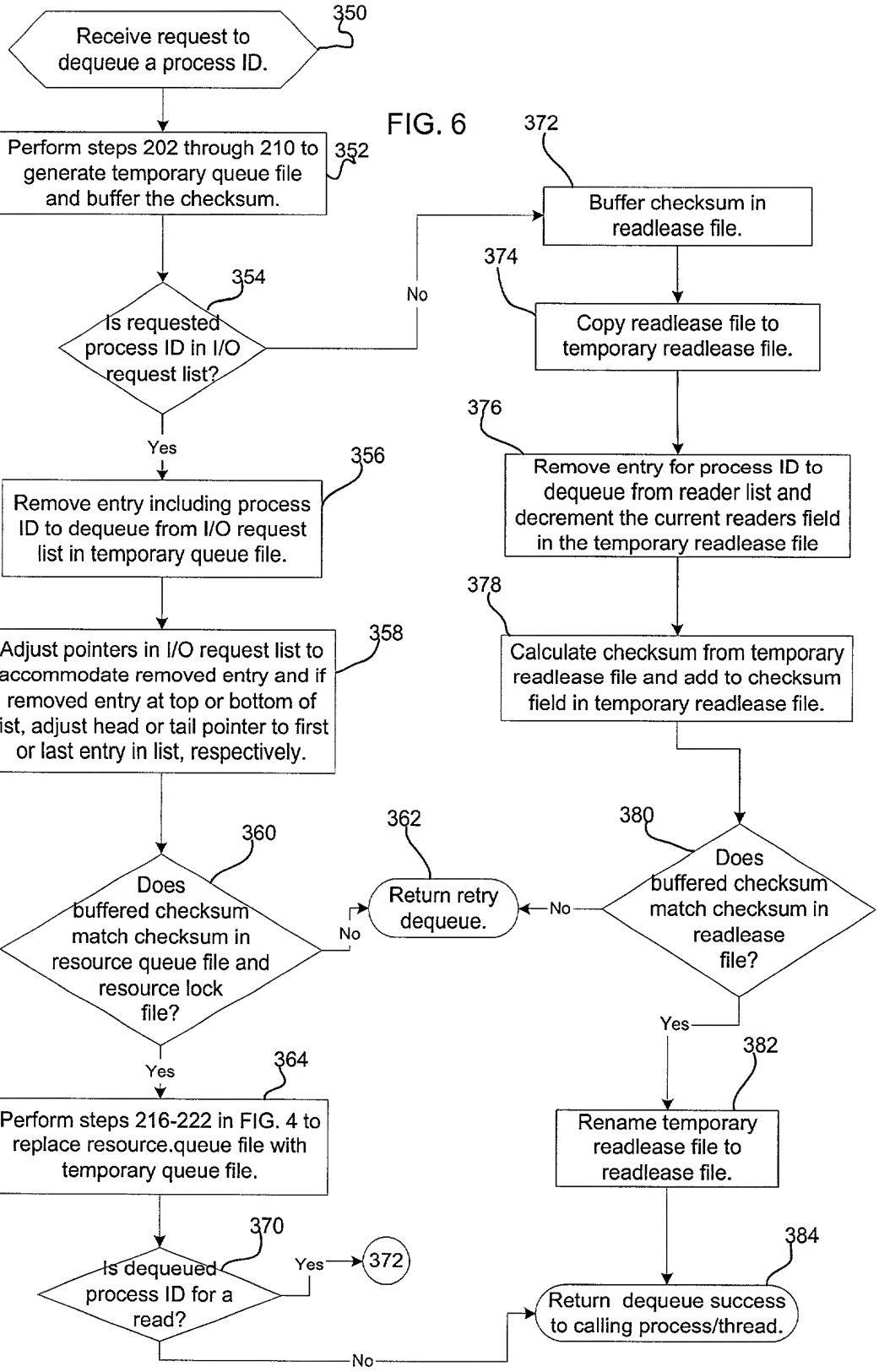
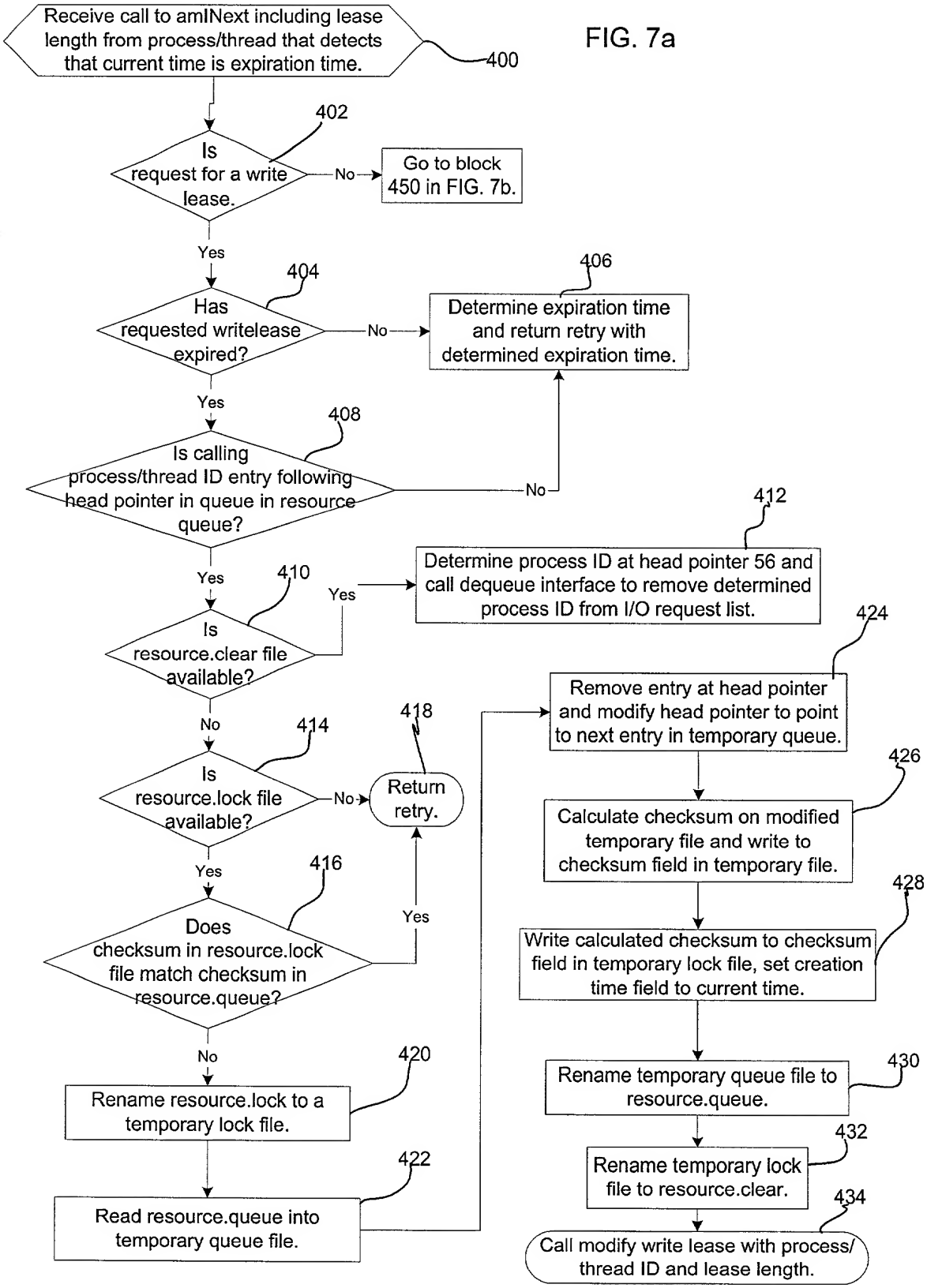

Receive request to dequeue a process ID.
350
Perform steps 202 through 210 to generate temporary queue file and buffer the checksum.
352
Is requested process ID in I/O request list?
354
No
Yes
Remove entry including process ID to dequeue from I/O request list in temporary queue file.
356
Adjust pointers in I/O request list to accommodate removed entry and if removed entry at top or bottom of list, adjust head or tail pointer to first or last entry in list, respectively.
358
Does buffered checksum match checksum in resource queue file and resource lock file?
360
No
Yes
Perform steps 216-222 in FIG. 4 to replace resource.queue file with temporary queue file.
364
Is dequeued process ID for a read?
370
Yes
372
No
Buffer checksum in readlease file.
372
Copy readlease file to temporary readlease file.
374
Remove entry for process ID to dequeue from reader list and decrement the current readers field in the temporary readlease file
376
Calculate checksum from temporary readlease file and add to checksum field in temporary readlease file.
378
Does buffered checksum match checksum in readlease file?
380
No
Return retry dequeue.
362
Yes
Rename temporary readlease file to readlease file.
382
Return dequeue success to calling process/thread.
384

FIG. 6

FIG. 7a


Receive call to amINext including lease length from process/thread that detects that current time is expiration time.
400
402
Is request for a write lease.
No
Go to block 450 in FIG. 7b.
Yes
404
Has requested writelease expired?
No
406
Determine expiration time and return retry with determined expiration time.
Yes
408
Is calling process/thread ID entry following head pointer in queue in resource queue?
No
412
Determine process ID at head pointer 56 and call dequeue interface to remove determined process ID from I/O request list.
Yes
410
Is resource.clear file available?
Yes
No
414
Is resource.lock file available?
No
418
Return retry.
Yes
416
Does checksum in resource.lock file match checksum in resource.queue?
Yes
No
420
Rename resource.lock to a temporary lock file.
422
Read resource.queue into temporary queue file.
424
Remove entry at head pointer and modify head pointer to point to next entry in temporary queue.
426
Calculate checksum on modified temporary file and write to checksum field in temporary file.
428
Write calculated checksum to checksum field in temporary lock file, set creation time field to current time.
430
Rename temporary queue file to resource.queue.
432
Rename temporary lock file to resource.clear.
434
Call modify write lease with process/ thread ID and lease length.

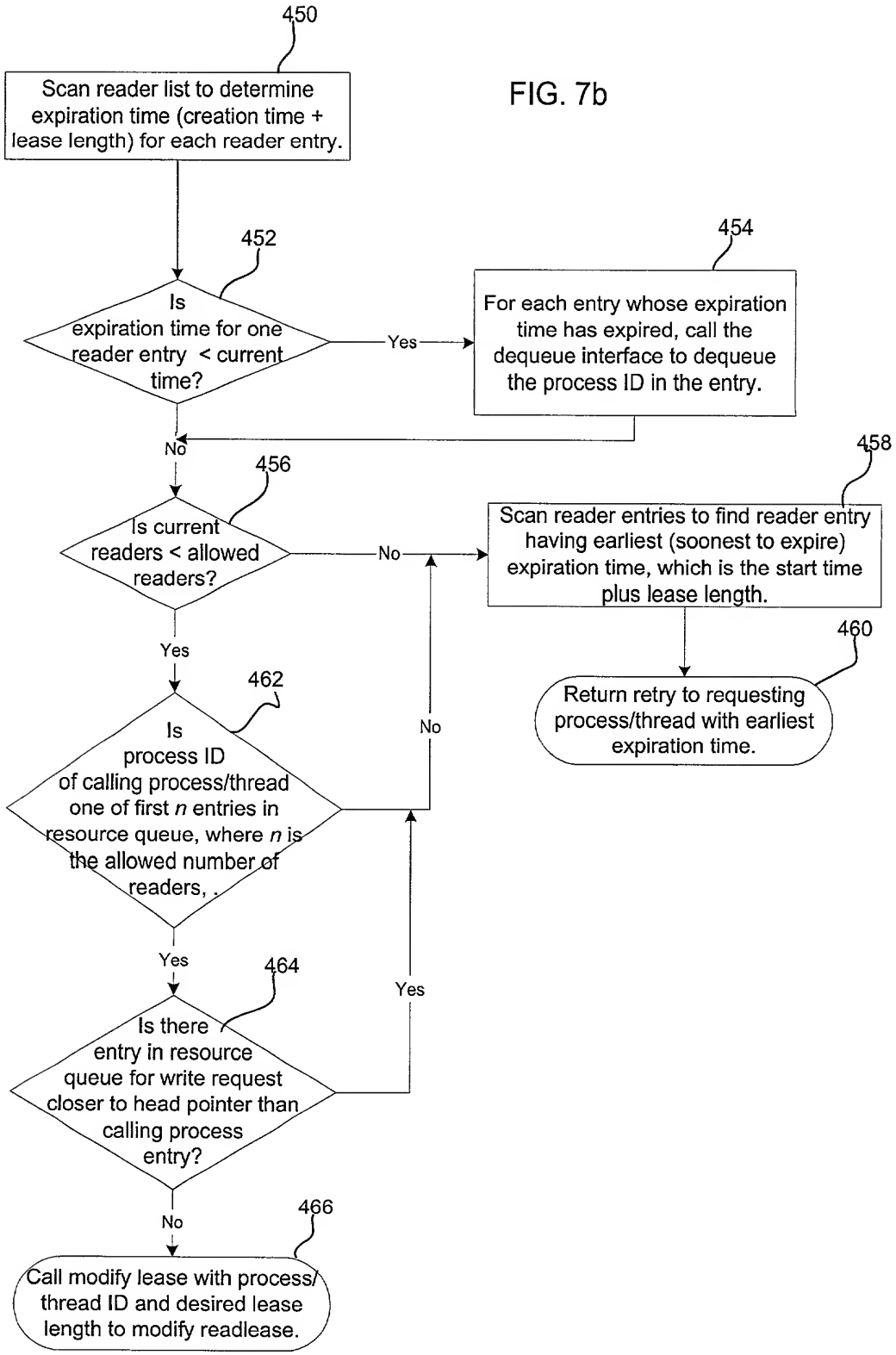

450
Scan reader list to determine expiration time (creation time + lease length) for each reader entry.
452
Is expiration time for one reader entry < current time?
Yes
454
For each entry whose expiration time has expired, call the dequeue interface to dequeue the process ID in the entry.
No
456
Is current readers < allowed readers?
No
458
Scan reader entries to find reader entry having earliest (soonest to expire) expiration time, which is the start time plus lease length.
460
Return retry to requesting process/thread with earliest expiration time.
Yes
462
Is process ID of calling process/thread one of first n entries in resource queue, where n is the allowed number of readers, .
No
Yes
464
Is there entry in resource queue for write request closer to head pointer than calling process entry?
Yes
No
466
Call modify lease with process/thread ID and desired lease length to modify readlease.

FIG. 7b

FIG. 8

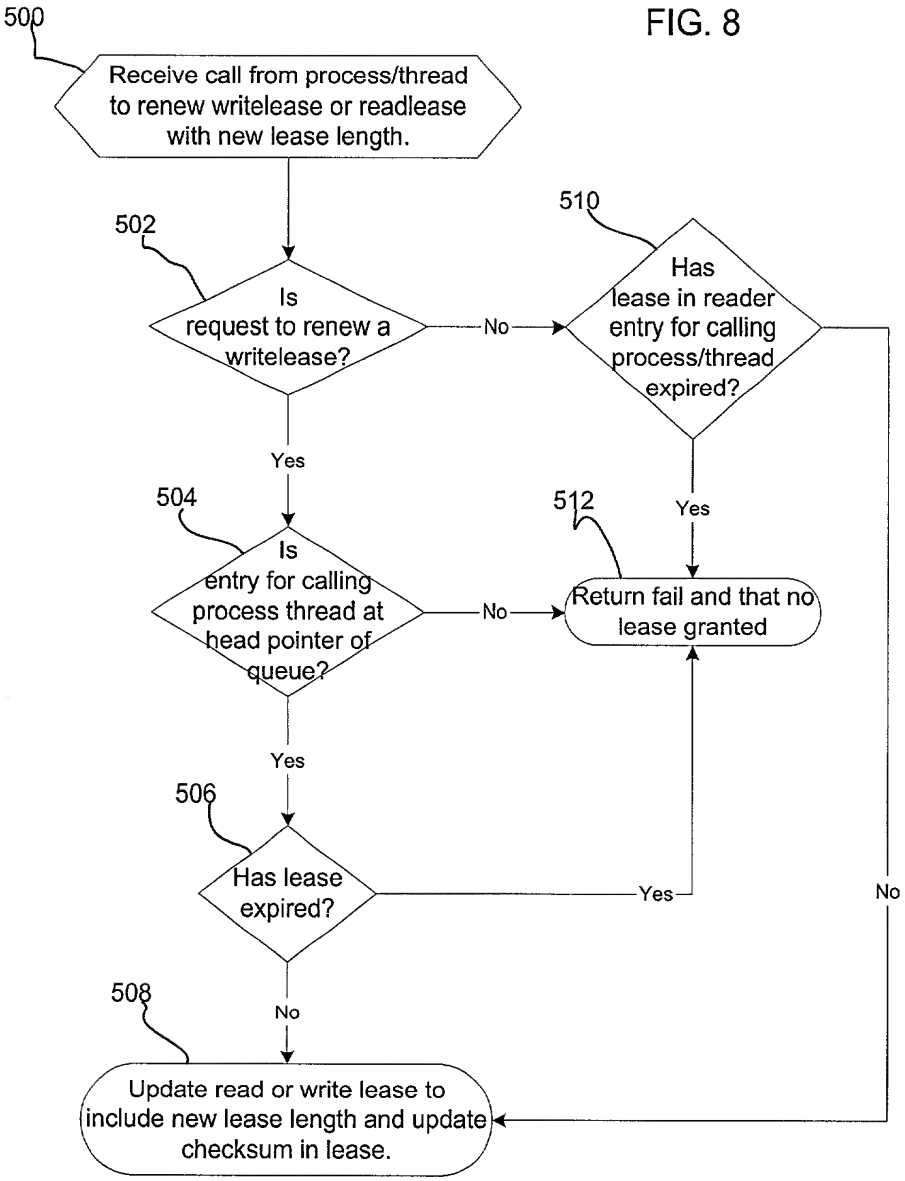
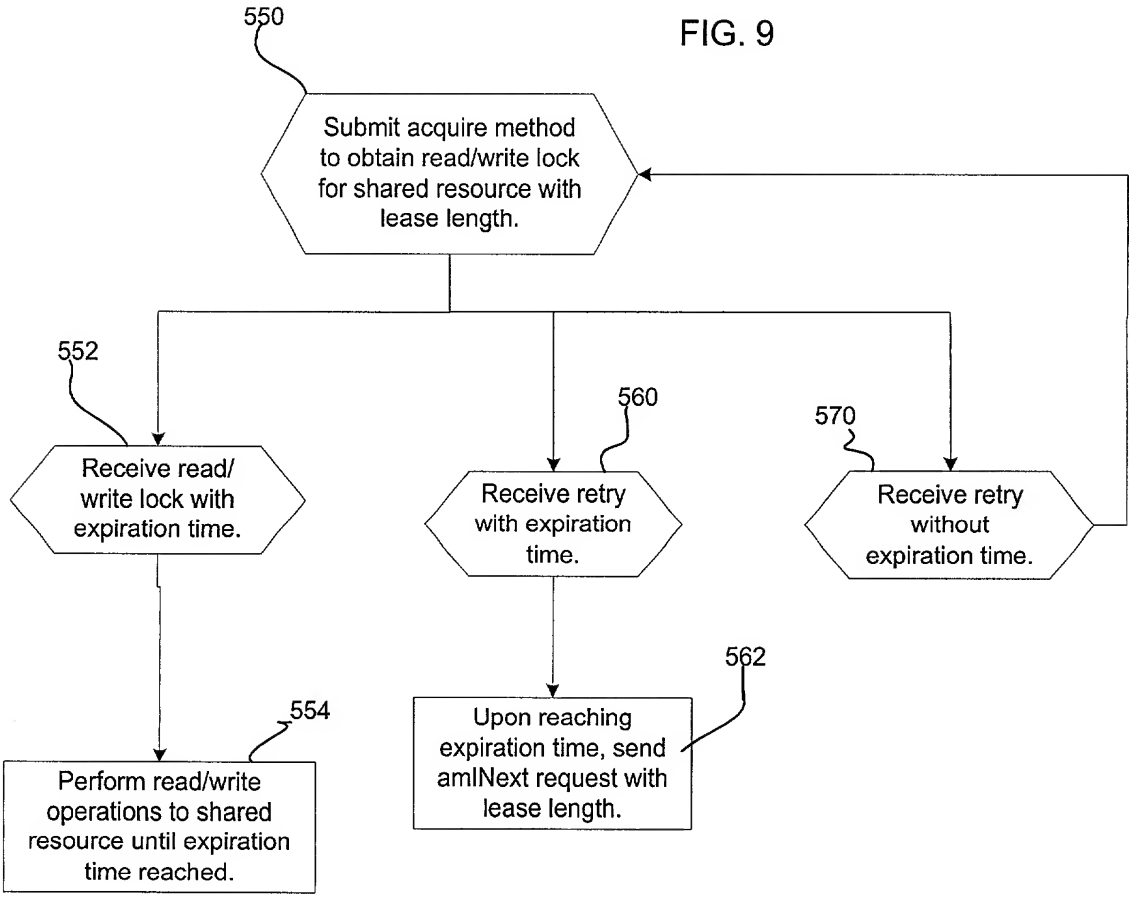

500
Receive call from process/thread to renew writelease or readlease with new lease length.
502
Is request to renew a writelease?
No
510
Has lease in reader entry for calling process/thread expired?
Yes
Yes
504
Is entry for calling process thread at head pointer of queue?
No
512
Return fail and that no lease granted
Yes
506
Has lease expired?
Yes
No
No
508
Update read or write lease to include new lease length and update checksum in lease.

FIG. 9


550
Submit acquire method to obtain read/write lock for shared resource with lease length.
552
Receive read/ write lock with expiration time.
560
Receive retry with expiration time.
570
Receive retry without expiration time.
554
Perform read/write operations to shared resource until expiration time reached.
562
Upon reaching expiration time, send amINext request with lease length.